**CS 1410: 15 Puzzle**

Graphical user interfaces, or GUIs, are a critical part of any modern application, including apps that run on phones, tablets, and televisions. Object-oriented programming plays a major role in developing graphical user interfaces, regardless of what programming language or technology is used. Using the object-oriented principles that we’ve learned, and by taking advantage of some already-existing tools, we can easily create programs with graphical user interfaces using Python.

**Assignment**

Your assignment is to create the [15 Puzzle](#) using Python and GTK. The assignment will consist of two sequential parts. For the second part, you are required to implement the logic which allows the game to be played according to its rules, which should include the following features and functionality:

1. If a tile button that is adjacent to the empty space is clicked, then that tile (along with its assigned number) should trade positions with the empty space. The empty space may have two, three, or four adjacent tiles, depending on its location in the grid. If any other non-adjacent tile button is clicked, then nothing should happen. This functionality alone should make the game playable, to a limited degree.

2. If the “New Game” button is clicked, then the tiles (as well as the empty space) should be reset to a random, scrambled configuration, in order to allow the player to begin arranging the tiles to solve the game.

For part 2 of the assignment, only the above functionality is required. You are welcome to continue working on additional features once you complete the requirements for part 2, but it is your responsibility to complete the requirements for part 2 of the assignment first, and submit it by the due date.

For this assignment, you should demonstrate the object-oriented principles that you have learned as you design and implement the classes that you will use to implement the graphical user interface and logic for the game. At a minimum, you should have separate classes for the user interface and game logic elements, and use aggregation to connect them.

**Extra Challenges**

- Create a game of a different size, such as 5 by 5, or allow the player to choose the size he or she wishes to play.

- Rather than using numbers to identify the individual tiles, display a rectangular division of a larger image inside each button. The objective of the game would then be to arrange the tiles in order to reveal the original picture.

**Hints**

- When swapping the tiles and their positions, do not attempt to alter the positions of the button objects within the containing grid. This is significantly more complicated than preserving the structure of the buttons and simply modifying their labels and visibility instead.

- Refer to the class examples to see examples of how to use the various GTK widget classes and methods.

- Refer to the Python GTK+ API documentation and tutorial to understand the various classes and methods that are available to use.

- When creating colors, use a helpful tool to determine the RGB values. Here are two good options: color.adobe.com and colorpicker.com

**Sample**

An example running program: